CLAIMS

What is claimed is:

1. A method for managing and controlling allocation and de-allocation of resources based on a guaranteed amount of resource and additional resources based on a best effort for a plurality of customers, said method comprising:

dynamically allocating server resources for a plurality of customers, such that said resources received by a customer are dynamically controlled and said customer receives a guaranteed minimum amount of resources as specified under a service level agreement (SLA).

2. The method according to claim 1, further comprising:

utilizing a performance metric to increase or decease an inbound traffic to a customer.

3. The method according to claim 1, further comprising:

supporting minimum and maximum server resource-based service level agreements for a plurality of customers.

4. The method according to claim 1, further comprising:

utilizing performance metrics to control the allocation of additional server resources to a plurality of customers using bounds on given service level metrics.

5. The method according to claim 1, further comprising:

supporting a plurality of service level metrics.

6. The method according to claim 1, further comprising:

selectively utilizing a plurality of different metrics for a plurality of different customers.

7. The method according to claim 1, further comprising:

utilizing a service level metric, an amount of allocable resources, and an inbound traffic rate, for defining a state of a current service level (M,N,R) for each customer.

8. The method according to claim 1, further comprising:

utilizing a target service level metric Mt to maintain an actual service level M substantially at or near a target service level so as to be guaranteed to fall between low and high bounds (Mlowbound and Mhighbound) specified in a service level agreement (SLA).

9. The method according to claim 1, further comprising:

computing a target amount of resources Nt and an inbound traffic rate Rt from a given target service level metric Mt and (M,N,R).

10. The method according to claim 1, further comprising:

performing at least one of a numerical analysis, a mathematical formulaic operation, an add-one/subtract-one, and a quick simulation for deriving a target amount of resources Nt and an inbound traffic rate Rt.

11. The method according to claim 1, further comprising:

supporting a resource utilization U for an actual service level M, average response time T for an actual service level M, and a response time percentile T% for an actual service level M, thereby to support targets of Ut, Tt and Tt%.

12. The method according to claim 1, further comprising:

deciding whether or not to add a server resource or to reduce an inbound traffic rate to meet service level agreements for a plurality of customers.

13. The method according to claim 1, further comprising:

providing a server farm including means for dynamically allocating servers or server resources to customers as demands of said customers change.

14. The method according to claim 1, further comprising:

designating a service level agreement (SLA) on a server resource for a customer as a form (Smin#(i), Smax#(i), Mbounds(i)), where Smin#(i) denotes a guaranteed minimum amount of server resources, Smax(i) denotes an upper bound on an amount of server resources that a customer desires to obtain when free resources are available, and Mbounds(i) that includes a low bound (Mlowbound(i)) and a high bound (Mhighbound(i)) designating bounds on a service level metric for allocating resources beyond the minimum amount Smin#(i) for each i- th customer.

15. The method according to claim 14, wherein a minimum amount of server resources Smin#(i) comprises a guaranteed amount of server resources that the i-th customer will receive regardless of the server resource usage, and

wherein a maximum amount of server resources Smax#(i) comprises the upper bound on the amount of server resources that the i-th customer may receive beyond the minimum amount provided that some unused server resources are available for allocation.

16. The method according to claim 15, wherein a range between Smin#(i) and Smax#(i) represents server resources that are provided on an as-available basis, such that the customer is not guaranteed to obtain these resources at any one time, if at all.

17. The method according to claim 1, wherein an allocation of an additional resource is performed so as to keep the performance metric within Mbounds(i).

18. The method according to claim 17, wherein said Mbounds(i) includes any one of bounds on the server resource utilization that are denoted by Ubounds(i), bounds on the average server response time that are denoted by Tbounds(i), and bounds on the server response time percentile that are denoted by T%bounds(i).

19. The method according to claim 1, further comprising:

when a server resource utilization goes above a predetermined set limit Mhighbound(i), attempting, by a server farm, to maintain the utilization between said predetermined set limits Mbounds(i) by allocating additional server resources to the i-th customer when free resources are available.

20. The method according to claim 19, further comprising:

if free resources are not available, then limiting, by the server farm, an amount of incoming traffic to the i-th customer's server.

21. The method according to claim 1, further comprising:

controlling said dynamic resource allocation to said plurality of customers to meet a value between the minimum and maximum server resources and performance metric-based service level agreements.

22. The method according to claim 1, further comprising:

monitoring an inbound traffic rate R(i), a currently assigned amount of server resources N(i), and a current service level metric M(i) for all of said plurality of customers.

23. The method according to claim 22, further comprising:

computing a target amount of server resources Nt(i), without changing an inbound traffic R(i).

24. The method according to claim 23, further comprising:

computing a target inbound traffic rate Rt(i), without changing an allocated resource N(i), to bring the service level metric M(i) to the targeted service level metric Mt(i) from monitored R(i), N(i) and M(i) for all i,

wherein the target service level metric Mt(i) comprises the service level metric substantially at or near where M(i) is to be maintained, and bounded by Mbounds(i).

25. The method according to claim 24, further comprising:

determining how to adjust a current M(i) to the target Mt(i), by one of changing N(i) to Nt(i) and by bounding the inbound traffic rate R(i) to Rt(i).

26. The method according to claim 25, further comprising:

requesting a system resource manager to perform the resource allocation.

27. The method according to claim 26, further comprising:

requesting an inbound traffic controller to throttle an amount of inbound traffic to the plurality of customers.

28. The method according to claim 1, further comprising:

maximizing revenue potential when allocating resources beyond a minimum amount for a customer.

29. The method according to claim 1, wherein a unit of said resources comprises a fixed size unit of allocable or de-allocable resources.

30. The method according to claim 1, wherein a unit of each allocable resource has a different amount.

31. A method of deciding server resource allocation for a plurality of customers, comprising:

computing target values (Nt(i),Rt(i)) for every customer i and setting a variable "ITC-informed(i)" = "no" for all customers "i" such that a record is kept of whether or not throttling on inbound traffic is being applied or not during a given service cycle time;

determining whether or not the service cycle time has expired;

if the service cycle time has not expired, then checking whether an operation state M(i) is within a predetermined area defined by a metric and a number of resources;

if the operation state is not within the predetermined area, then checking whether any customer exists such that a target resource amount Nt(i) is less than a current amount N(i);

if Nt(i) is less than N(i), then determining whether the inbound traffic has been throttled, and determining whether any "i" is ITC-informed(i); and

if the inbound traffic has been throttled, then removing the throttling by directing an inbound traffic controller to stop throttling i-th traffic class and setting ITC-informed (i) = "no".

32. The method according to claim 31, further comprising:

when Nt(i) is less than N(i) and it is determined that the inbound traffic is not throttled, deallocating resources from said customers.

33. The method according to claim 32, further comprising:

determining whether the resources must be increased by selecting any i and determining whether Nt(i) is greater than N(i).

34. The method according to claim 33, further comprising:

if it is determined that Nt(i) is greater than N(i) and if free resources are judged to be available, then allocating additional resources up to Nt(i) - N(i) resources without exceeding a maximum amount of server resources Smax#(i)).

35. The method according to claim 33, further comprising:

if it is determined that Nt(i) is greater than N(i) and if free resources are judged to be unavailable and if the currently allocated resource N(i) is less than the guaranteed minimum Smin#(i), then reclaiming resources from those customers j having more than a guaranteed minimum such that N(j) > Smin#(j).

36. The method according to claim 33, further comprising:

if it is determined that Nt(i) is greater than N(i) and if free resources are judged to be unavailable and if the currently allocated resource N(i) is more than or equal to the guaranteed minimum Smin#(i), then throttling the inbound traffic.

37. The method according to claim 36, further comprising:

bounding, by the inbound traffic controller, the traffic by Rt(i).

38. The method according to claim 31, further comprising:

searching for a potential revenue maximization opportunity when allocating free resources to various customers.

39. The method according to claim 38, further comprising:

first seeking to de-allocate resources, then allocating additional resources to customers whose service level metric is outside of a predetermined area, and thirdly searching for when the customer's inbound traffic must be throttled due to exhaustion of free resources or the maximum amount of resources has been already allocated.

40. A system for managing and controlling allocation and de-allocation of resources based on a guaranteed amount of resources and additional resources based on a best effort for a plurality of customers, said system comprising:

a plurality of servers; and

a resource allocation device for dynamically allocating server resources for a plurality of customers, such that said resources received by a customer are dynamically controlled and said customer receives a guaranteed minimum amount of resources as specified under a service level agreement (SLA).

41. A system for managing server resources for a plurality of customers, comprising:

a main system;

an inbound traffic controller operably coupled to said main system; and

a server resource manager coupled to said main system,

wherein said main system includes a decision module, a module for computing a target amount of resources Nt(i) and a target inbound traffic rate Rt(i), and a repository for storing Service Level Agreements,

wherein said decision module computes the target values Nt(i) and Rt(i) from monitored service level data M(i), N(i) and R(i) for every customer, such that a resource allocation is dynamically optimized for each customer.

42. The system according to claim 41, wherein an allocation and de-allocation of said resources is based on a guaranteed amount of resource and additional resources based on a best effort for the plurality of customers.

43. The system according to claim 41, wherein said resources are dynamically allocated for the plurality of customers, such that said resources received by a customer are dynamically controlled and said customer receives a guaranteed minimum amount of resources as specified under a service level agreement (SLA).

44. The system according to claim 41, wherein said decision module, based on the SLA information, (M(i),N(i),R(i)), Nt(i) and Rt(i), decides which action to take, to reallocate resources.

45. The system according to claim 44, wherein the decision module decides one of changing the current resource amount from N(i) to the target resource amount Nt(i), and bounding a current inbound traffic rate R(i) by Rt(i).

46. The system according to claim 45, wherein said main system instructs said server resource manager to change resource allocation and for instructing said inbound traffic controller to bound the incoming traffic to a specific customer site.

47. A program product device for storing a program for execution by a digital data processing apparatus to perform a method of managing and controlling allocation and de-allocation of resources based on a guaranteed amount of resource and additional resources based on a best effort for a plurality of customers, said method comprising:

dynamically allocating server resources for a plurality of customers, such that said resources received by a customer are dynamically controlled and said customer receives a guaranteed minimum amount of resources as specified under a service level agreement (SLA).

48. A program product device for storing a program for execution by a digital data processing apparatus to perform a method of deciding server resource allocation for a plurality of customers, comprising:

computing target values (Nt(i),Rt(i)) for every customer i and setting a variable "ITC-informed(i)" = "no" for all customers "i" such that a record is kept of whether or not throttling on inbound traffic is being applied or not during a given service cycle time;

determining whether or not the service cycle time has expired;

if the service cycle time has not expired, then checking whether an operation state M(i) is within a predetermined area defined by a metric and a number of resources;

if the operation state is not within the predetermined area, then checking whether any customer exists such that a target resource amount Nt(i) is less than a current amount N(i);

if Nt(i) is less than N(i), then determining whether the inbound traffic has been throttled, and determining whether any "i" is ITC-informed(i); and

if the inbound traffic has been throttled, then removing the throttling by directing an inbound traffic controller to stop throttling i-th traffic class and setting ITC-informed (i) = "no".